



Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo

Compiladores: Autômatos finitos

Prof^a Thaína A. A. Tosta

tosta.thaina@unifesp.br

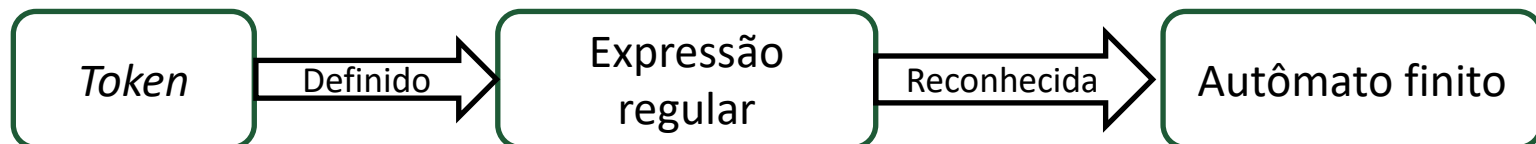
São José dos Campos – 2021/2

Autômatos finitos

- Um autômato finito é uma máquina de estados finitos que permite reconhecer a ocorrência de símbolos de um alfabeto e determinar se uma cadeia de caracteres (*string*) pertence ou não a uma linguagem regular;
- Os autômatos podem ser usados para descrever o processo de reconhecimento de cadeias de entrada (identificação de *tokens*);
- Portanto, são muito úteis na construção do **Analizador Léxico**.

Autômatos finitos

- O núcleo do analisador léxico é uma implementação de um autômato finito:
 - A máquina de estados finitos aceita símbolos de uma *string*;
 - Ao final da *string*, indica se ela é válida ou não;
- O autômato é definido para cada conjunto de símbolos que deve ser reconhecido;
- Cada *token* pode ser descrito por uma lista ou por uma expressão regular.



Autômatos finitos

Definição formal

Autômato é descrito por uma quintupla

$$M = (K, \Sigma, \delta, s, F)$$

K conjunto (finito) de estados

Σ alfabeto (finito) de entrada

δ conjunto de transições

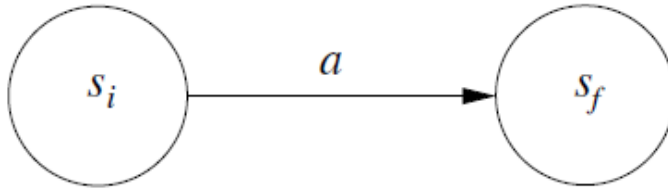
s estado inicial, $s \in K$

F conjunto de estados finais, $F \subseteq K$

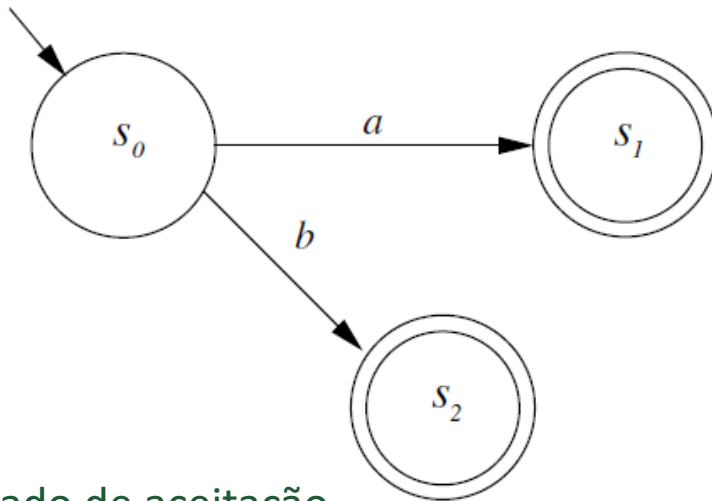
Autômatos finitos

Representação gráfica

Estados e transição



Estados inicial e finais



Representação Formal:

$$K = \{s_0, s_1, s_2\}$$

$$\Sigma = \{a, b\}$$

$$\delta = \{(s_0, \{a\}, s_1), (s_0, \{b\}, s_2)\}$$

$$s = s_0$$

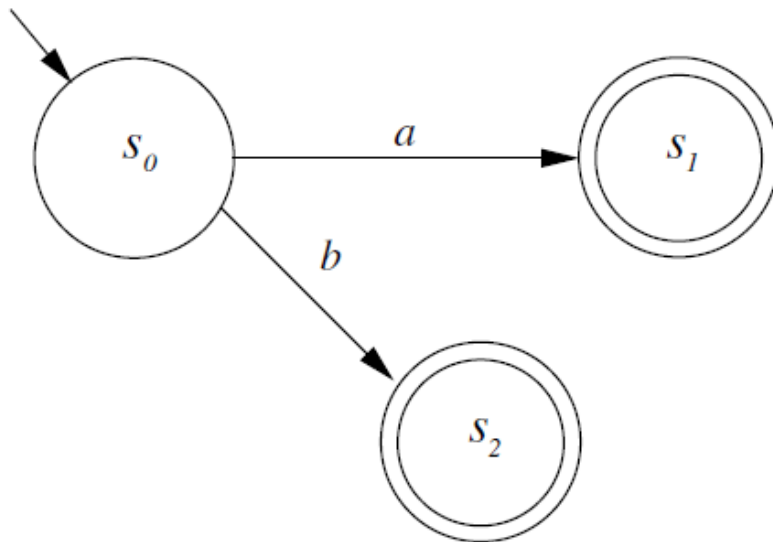
$$F = \{s_1, s_2\}$$

estado final = estado de aceitação
(um ou mais)

Autômatos finitos

Representação tabular

Estados inicial e finais



	s_0	s_1	s_2
a	s_1	—	—
b	s_2	—	—

Estado inicial: s_0

Estados finais: s_1 , s_2

Autômatos finitos

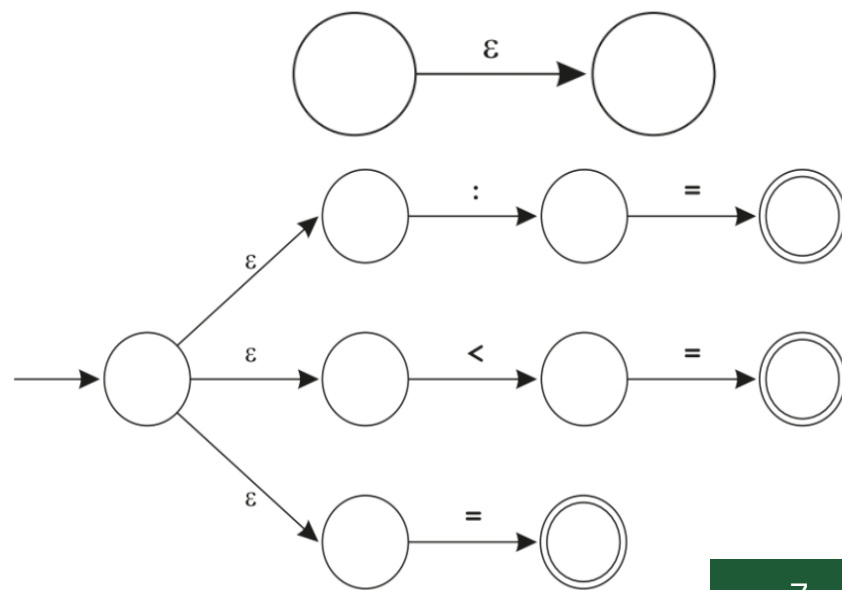
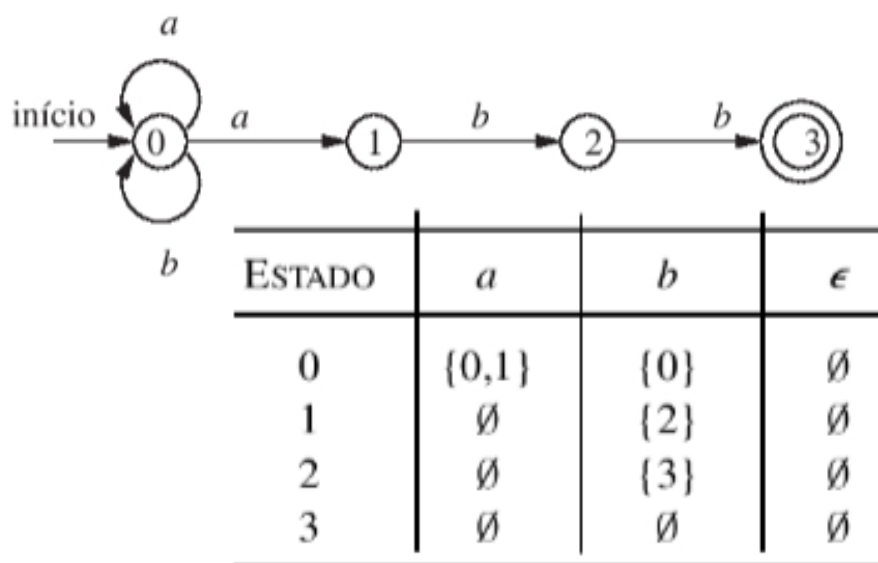
Os autômatos finitos dividem-se entre:

Não-Determinísticos (AFND ou NFA)	Determinísticos (AFD ou DFA)
Mais lentos	Mais rápidos
Menores	Maiores

Autômatos finitos

Não-Determinísticos

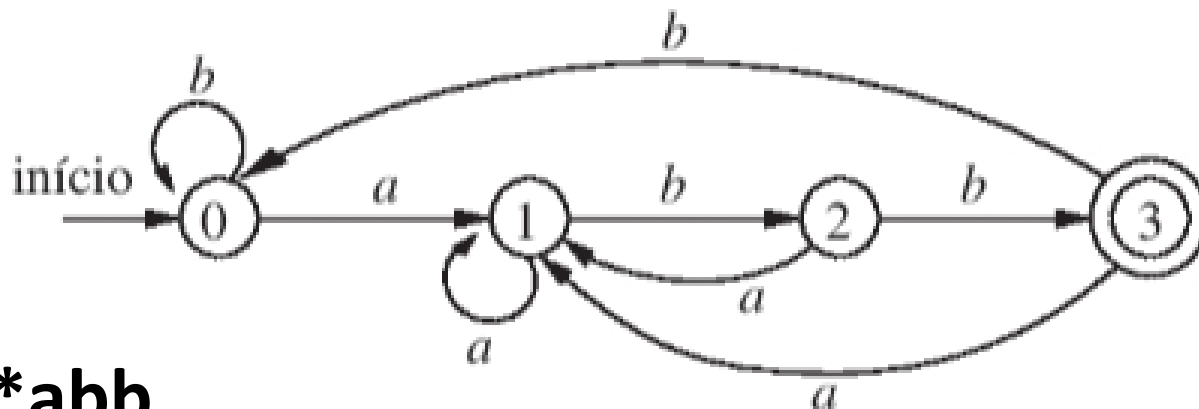
- Pode existir mais de uma transição (aresta) saindo do mesmo estado com o mesmo símbolo;
- Pode existir transição de estado sem a ocorrência de nenhum símbolo de entrada: a transição pela *string* vazia (anotada por ϵ).



Autômatos finitos

Determinísticos

- Não há alternativas para a transição a partir de um estado com o mesmo símbolo de entrada;
- Não há transições pela *string* vazia;
- O AFD é um caso especial de AFND.

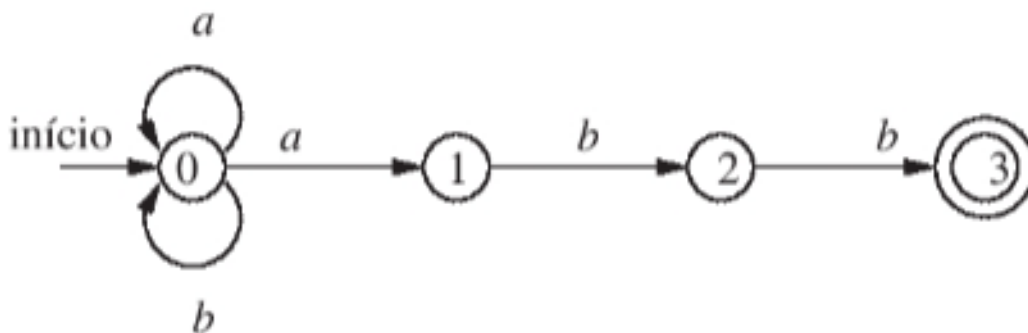


$(a | b)^*abb$

Autômatos finitos

Aceitação das cadeias de entrada pelos autômatos

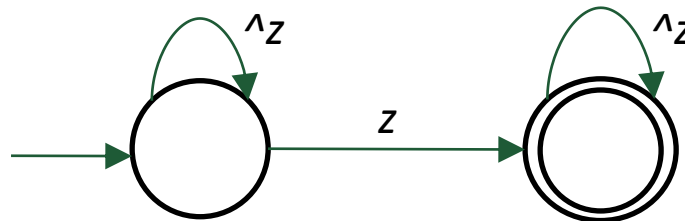
Um autômato finito **aceita** (consome) a cadeia de entrada **se e somente se** houver algum caminho no grafo de transições do estado inicial para um dos estados finais (**estados de aceitação**).



Autômatos finitos

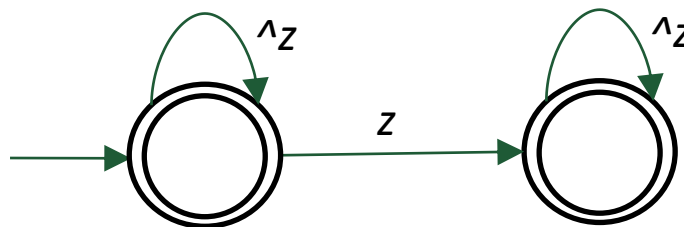
Exemplos de AFDs com $\Sigma = \{a, b, c, \dots, z\}$

- Reconhecimento de cadeias de caracteres que contêm **exatamente um z**



$$[^az]^*z[^az]^*$$

- Reconhecimento de cadeias de caracteres que contêm **no máximo um z**



$$[^az]^*[z|\varepsilon][^az]^* \\ \text{ou } [^az]^*z?[^az]^*$$

Autômatos finitos

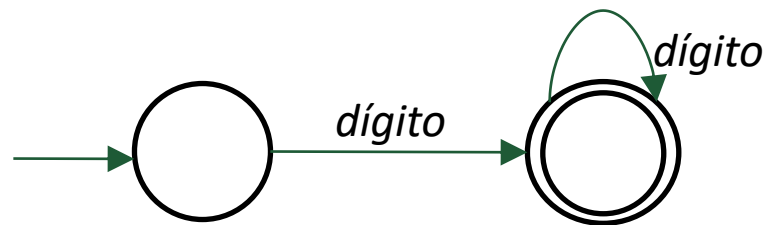
Exemplos de AFDs

- Reconhecimento de um número natural

Definições regulares:

$\text{dígito} = [0-9]$

$\text{nat} = \text{dígito}^+ (= \text{dígito} \text{dígito}^*)$



Autômatos finitos

Exemplos de AFDs

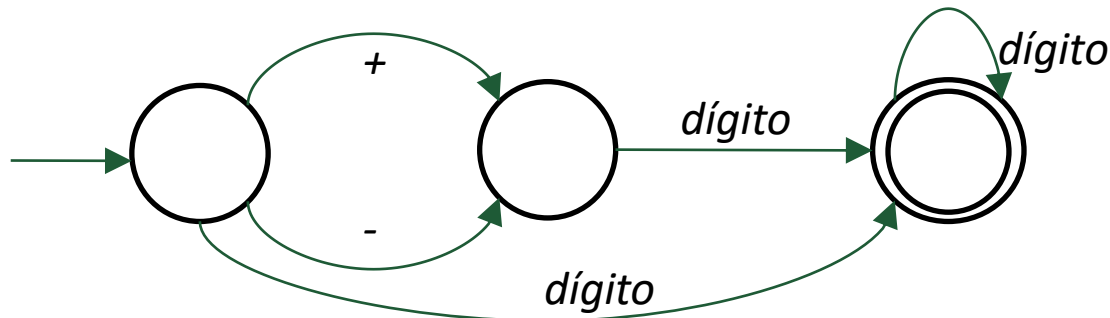
- Reconhecimento de um número natural com sinal opcional

Definições regulares:

dígito = [0-9]

nat = dígito+

signedNat = (+|-)? nat



Autômatos finitos

Exemplos de AFDs

- Reconhecimento de um número real

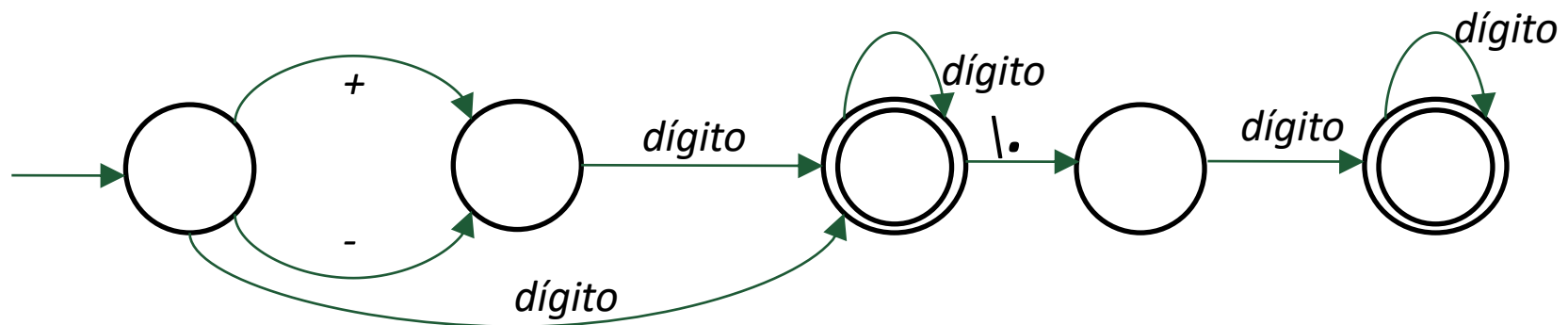
Definições regulares:

dígito = $[0-9]$

nat = dígito^+

signedNat = $(+|-)? \text{nat}$

real = $\text{signedNat} (\backslash.\text{nat})?$



Autômatos finitos

Exercício

- Reconhecimento de um número real com parte exponencial opcional

Definições regulares:

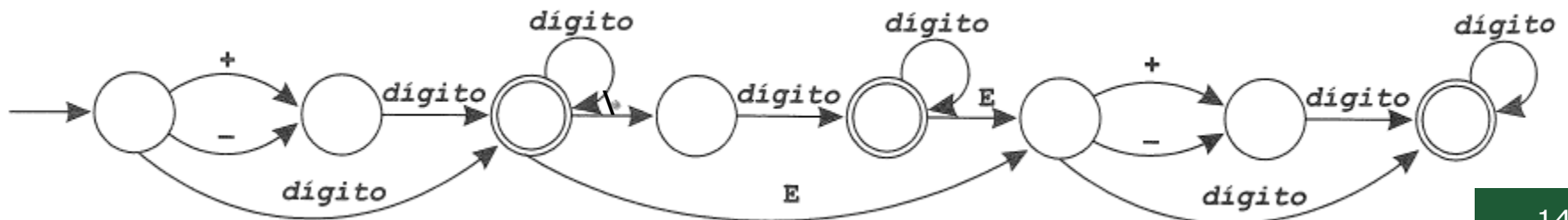
dígito = [0-9]

nat = dígito⁺

signedNat = (+|-)? nat

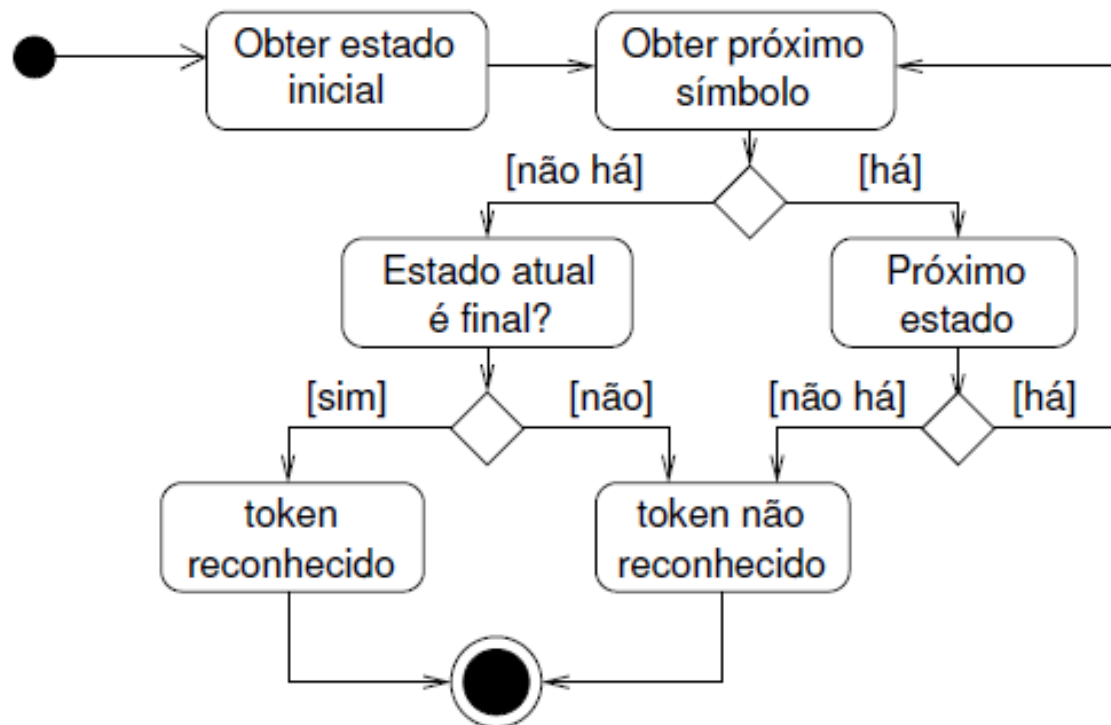
exp = E signedNat

real = signedNat (\.nat)? exp?



Autômatos finitos

Algoritmo para a identificação de cadeias feita por um AFD (notação de digrama de atividades da UML)



Autômatos finitos

Exercícios

Construa AFDs para as expressões regulares abaixo:

$a (a \mid b \mid c \mid \dots \mid z)^* a \mid a$

$(1 \mid 2 \mid 3 \mid \dots \mid 9) (0 \mid 1 \mid 2 \mid \dots \mid 9)^* \mid 0 \mid \varepsilon$

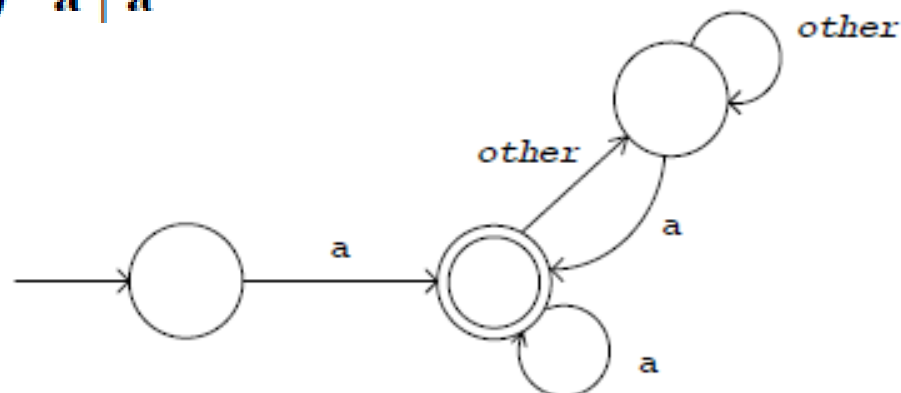
$(0 \mid 1 \mid 2 \mid \dots \mid 8)^* (0 \mid 1 \mid 3 \mid 4 \mid \dots \mid 9)^*$

$b^* a b^* (a b^* a b^*)^* \mid a^* b a^* (b a^* b a^*)^*$

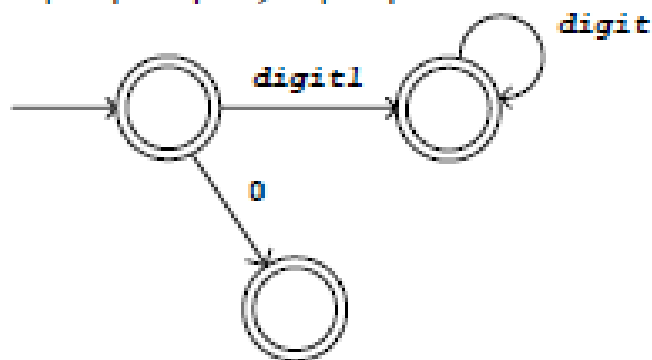
Autômatos finitos

$a (a \mid b \mid c \mid \dots \mid z)^* a \mid a$

other = $b \mid c \mid \dots \mid z$



$(1 \mid 2 \mid 3 \mid \dots \mid 9)(0 \mid 1 \mid 2 \mid \dots \mid 9)^* \mid 0 \mid \epsilon$

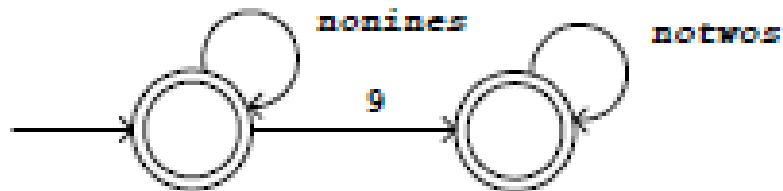


$(digit1 = 1 \mid 2 \mid 3 \mid \dots \mid 9)$

$(digit = 0 \mid 1 \mid \dots \mid 9)$

Autômatos finitos

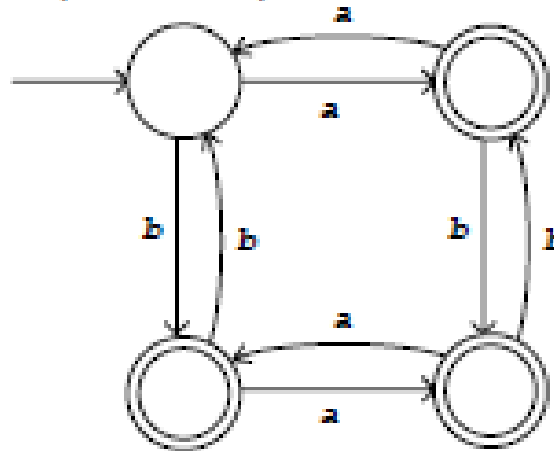
$(0 | 1 | 2 | \dots | 8)^* (0 | 1 | 3 | 4 | \dots | 9)^*$



$(nonines = 1 | 2 | 3 | \dots | 8)$

$(notwos = 0 | 1 | 3 | 4 | \dots | 9)$

$b^*ab^*(ab^*ab^*)^* | a^*ba^*(ba^*ba^*)^*$

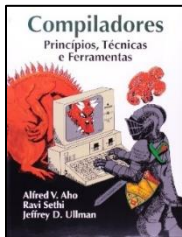


Autômatos finitos

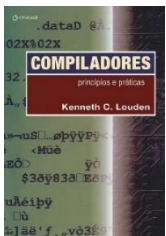
Bibliografia consultada



RICARTE, I. **Introdução à Compilação**. Rio de Janeiro: Editora Campus/Elsevier, 2008.



AHO, A. V.; LAM, M. S.; SETHI, R. e ULLMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. 2ª edição – São Paulo: Pearson Addison-Wesley, 2008.



LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004.